Documento de diseño proyecto biometría – GTI-3A

Contenido

[Idea General Proyecto 3](#_Toc83998804)

[Aplicación Arduino 3](#_Toc83998805)

[Aplicación Android 4](#_Toc83998806)

[Diseño Servidor 5](#_Toc83998807)

[Lógica Negocio Servidor 5](#_Toc83998808)

[Entidad Relación Base de Datos 6](#_Toc83998809)

[Diseño Cliente Web 6](#_Toc83998810)

Ilustraciones

[Ilustración 1 - Esquema General Proyecto 3](#_Toc83998828)

[Ilustración 2 - Diagrama de Clases Arduino 3](#_Toc83998829)

[Ilustración 3 - Diagrama de Interacción Loop Arduino 4](#_Toc83998830)

[Ilustración 4 - Diagrama de clases Android 4](#_Toc83998831)

[Ilustración 5 - Diagrama de Interacción Android Escuchar Beacon y Enviar Datos Al Servidor 5](#_Toc83998832)

[Ilustración 6 - Diagrama de clases Servidor y REST 5](#_Toc83998833)

[Ilustración 7 - Esquema Relacional BD 6](#_Toc83998834)

[Ilustración 8 - Diagrama de clases Cliente Web 6](#_Toc83998835)

[Ilustración 9 - Diagrama de Interacción Listar Mesuras en Cliente Web 7](#_Toc83998836)

# Idea General Proyecto

El conjunto de la aplicación consta de 4 partes principalmente. Un sensor que recoge datos de los gases del ambiente para que sean procesados y calcular la calidad del aire. Estos datos son enviados a través de bluetooth que serán recibidos por una terminal Android. La aplicación Android los procesa y los envía vía REST al servidor. Por último, hay una aplicación web que nos permite observar estos datos.

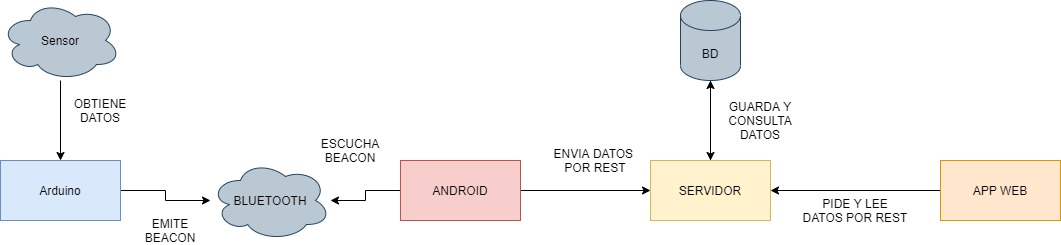


Ilustración - Esquema General Proyecto

# Aplicación Arduino

El siguiente diseño de código define el funcionamiento del microprocesador al que se le conecta un sensor de gases para detectar la calidad del aire. Los datos que recoge son enviados vía bluetooth (beacons de tipo IBeacon) para ser tratados posteriormente por un terminal Android.

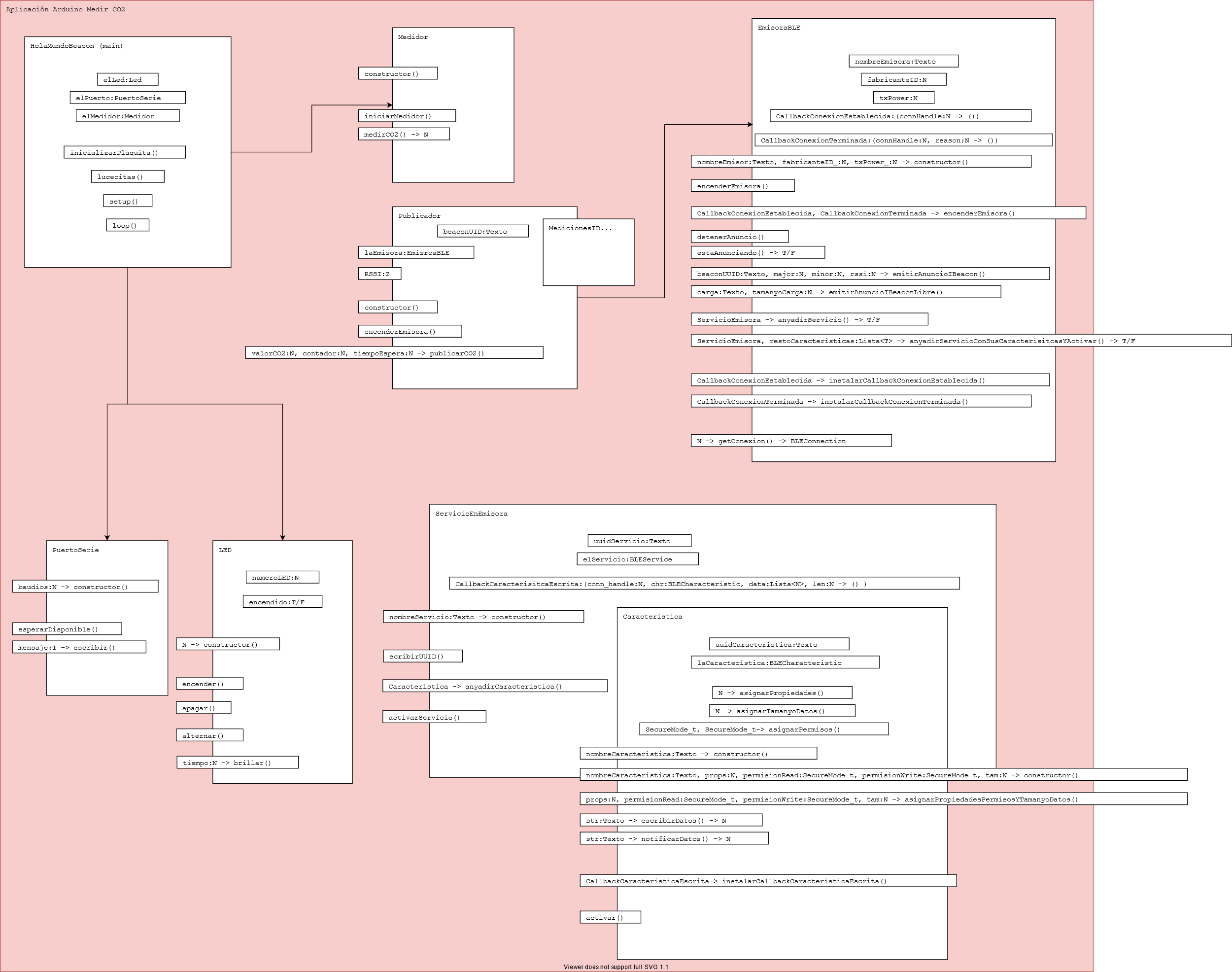


Ilustración - Diagrama de Clases Arduino

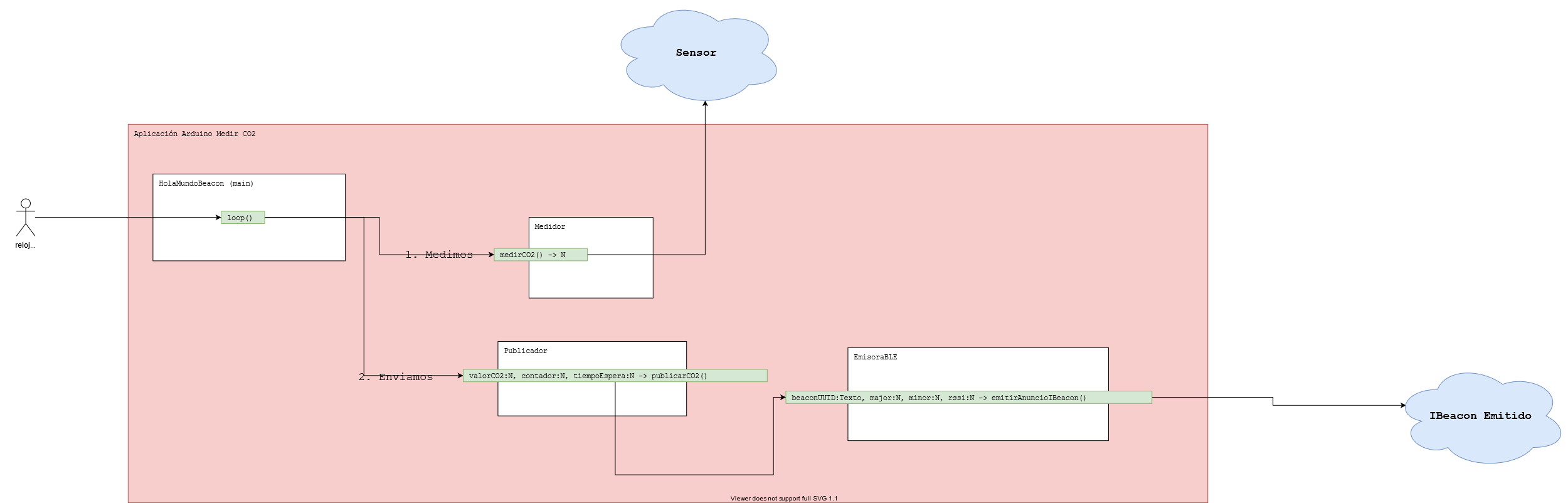


Ilustración - Diagrama de Interacción Loop Arduino

# Aplicación Android

El siguiente diseño explica como el usuario puede con una aplicación Android buscar su dispositivo con el sensor y enviar estos datos al servidor

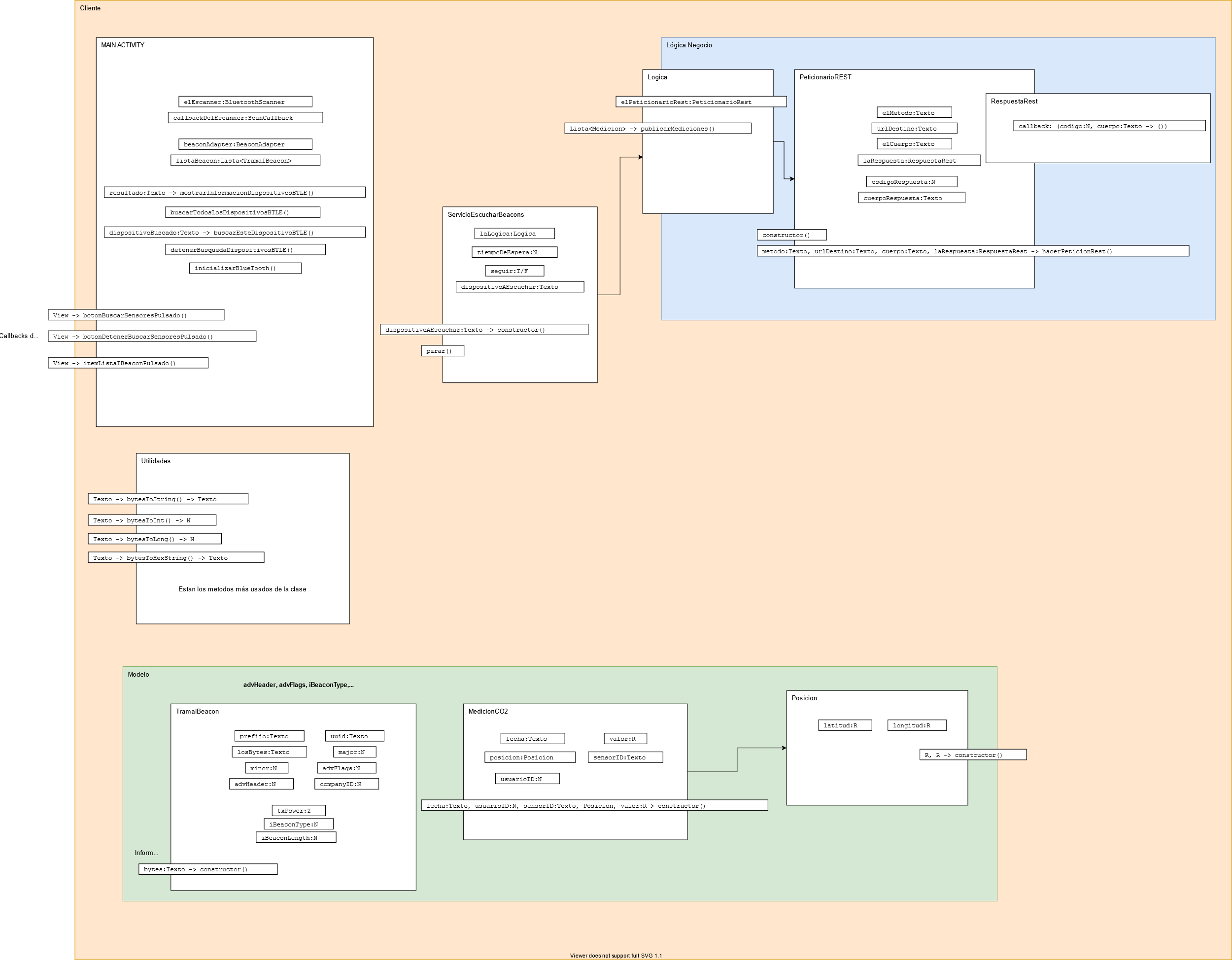


Ilustración - Diagrama de clases Android

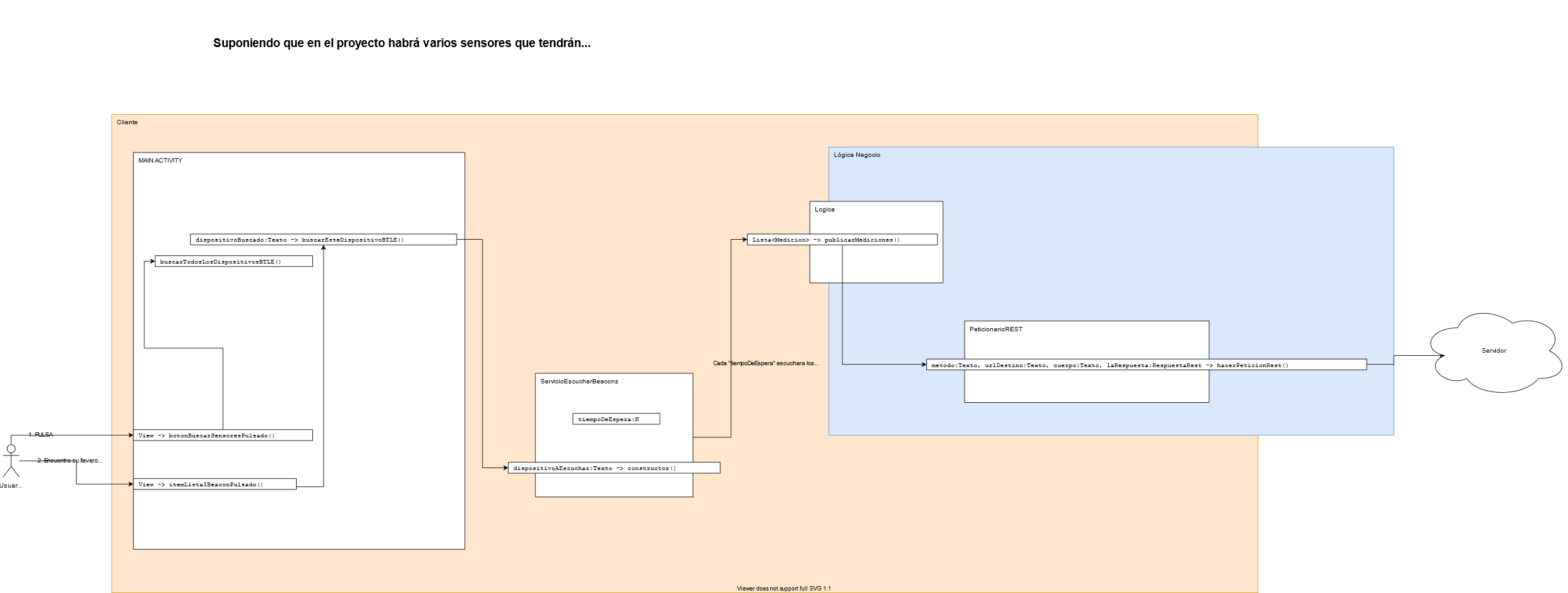


Ilustración - Diagrama de Interacción Android Escuchar Beacon y Enviar Datos Al Servidor

# Diseño Servidor

## Lógica Negocio Servidor

Aquí se define el diseño de la API REST del servidor y su lógica para interactuar con la Base de Datos

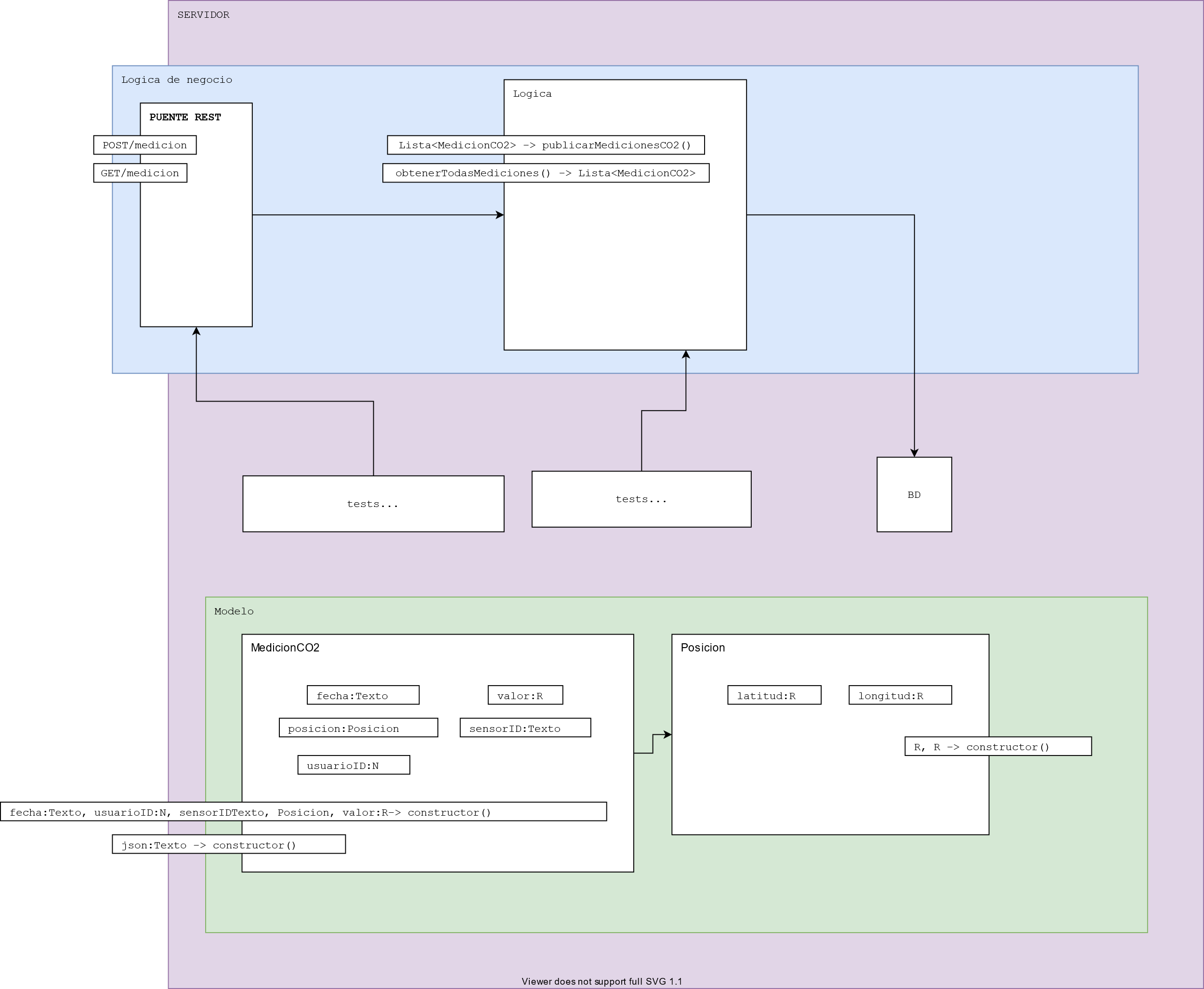


Ilustración - Diagrama de clases Servidor y REST

## Entidad Relación Base de Datos

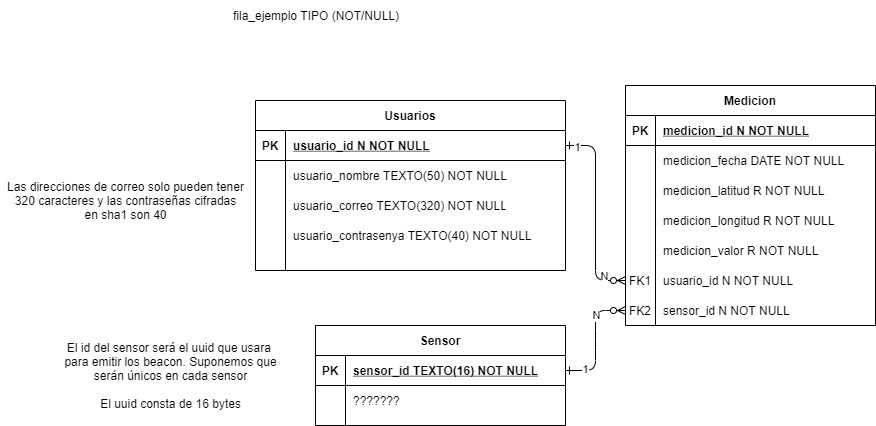


Ilustración - Esquema Relacional BD

# Diseño Cliente Web

El siguiente diseño define el cliente web del proyecto. Como el ux interactúa con una lógica “falsa” que es una replica de la del servidor (no en su totalidad) que esta a su vez interactúa con la API REST del servidor

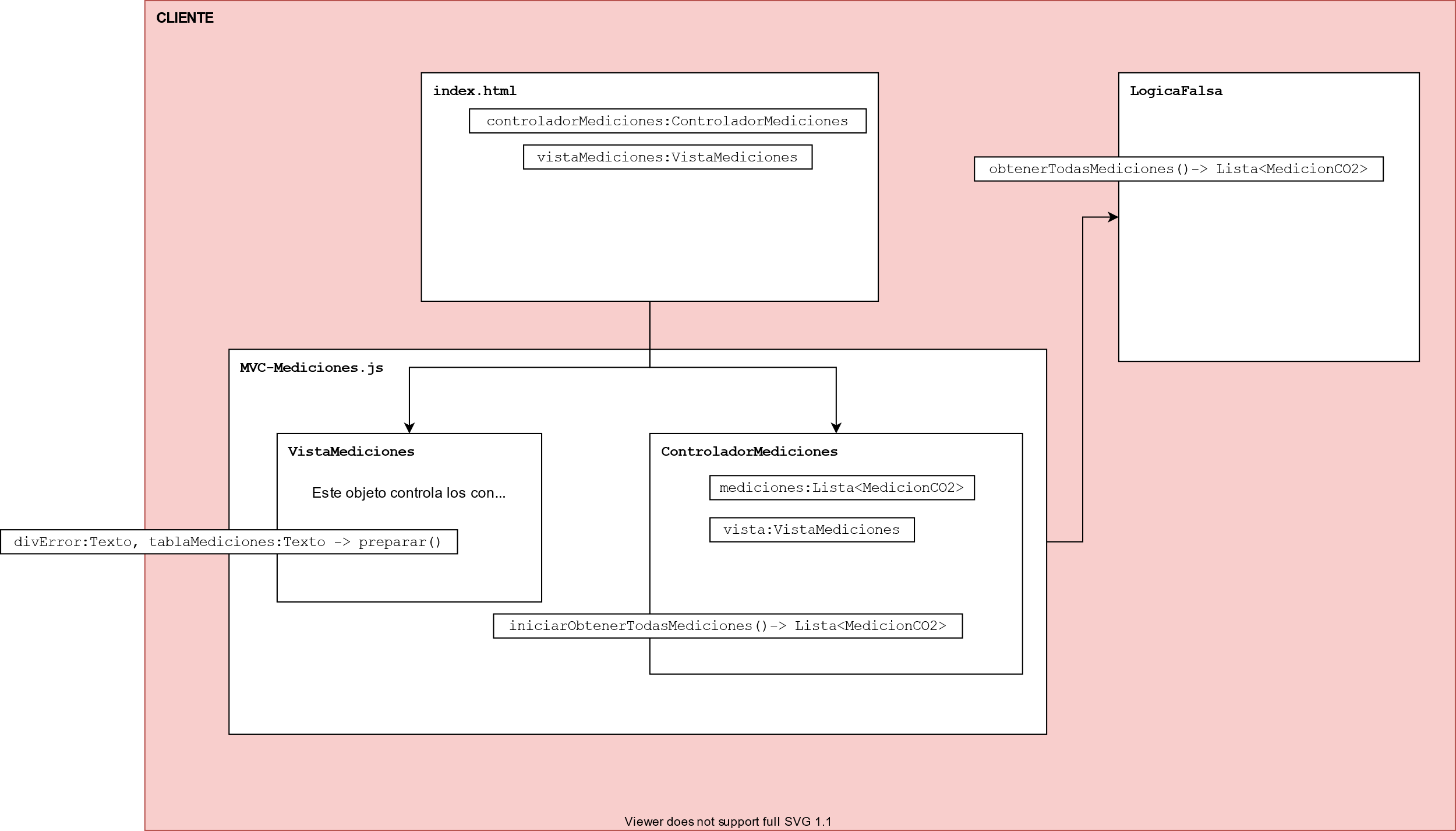


Ilustración - Diagrama de clases Cliente Web

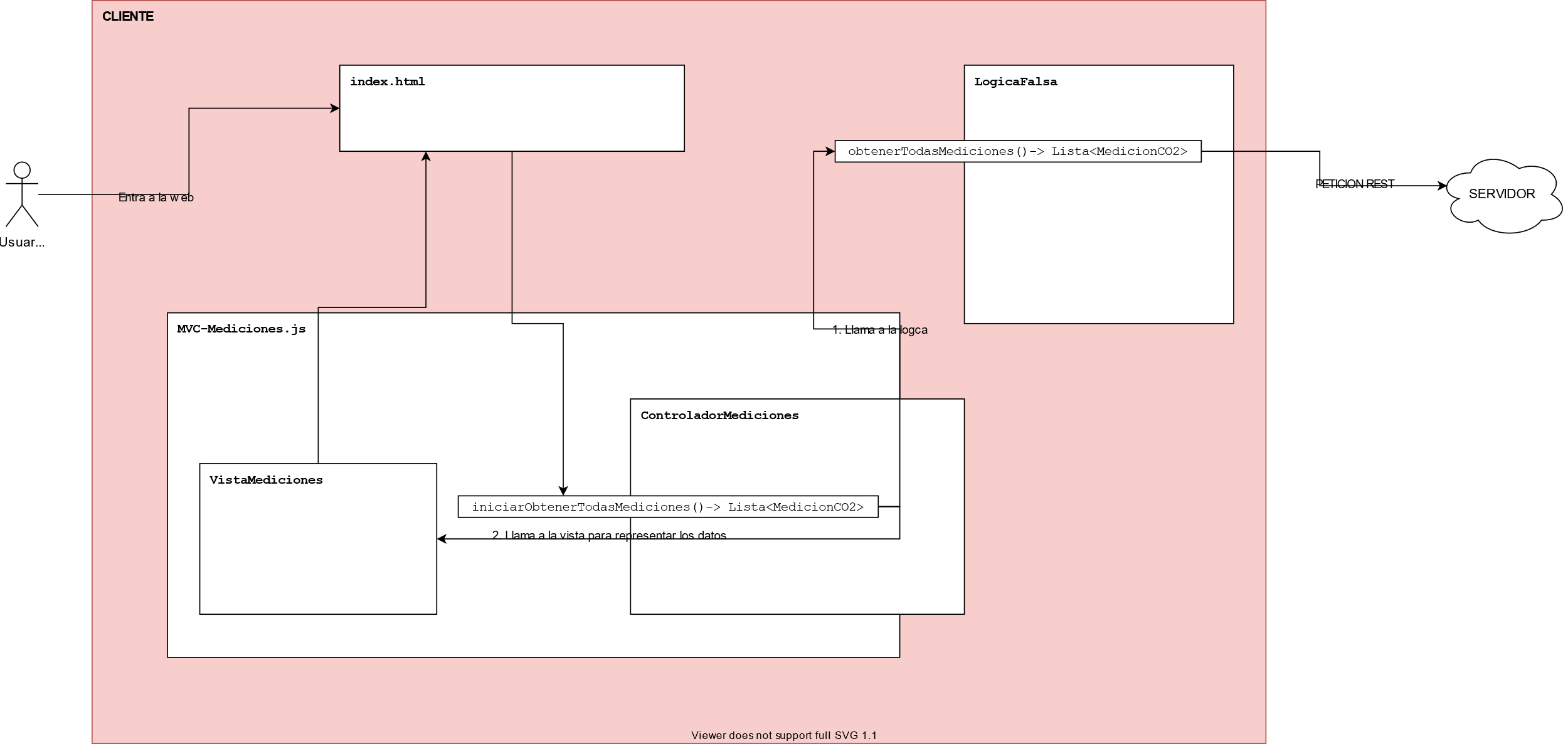


Ilustración - Diagrama de Interacción Listar Mesuras en Cliente Web